

# APLIKASI TRANSFORMASI HOUGH UNTUK DETEKSI GARIS LURUS

Halimatus Sa'diyah\*, R.Rizal Isnanto\*\*, Achmad Hidayatno\*\*

## Abstrak

Deteksi suatu garis lurus pada citra digital kadang dibutuhkan dalam aplikasi pengolahan citra berikutnya, misalnya menyambungkan garis-garis yang terputus ataupun mendeteksi bentuk objek. Untuk itulah dibuat program aplikasi transformasi Hough untuk deteksi garis lurus.

Bahasa pemrograman yang digunakan untuk membuat program dalam penelitian ini adalah Matlab 7.1. Penelitian dimulai dari pembacaan citra, pengubahan citra asli menjadi citra aras keabuan, deteksi tepi, transformasi citra tepi dari koordinat Kartesian ke dalam koordinat polar, deteksi dan rekonstruksi garis lurus.

Setelah dilakukan proses pengujian, diperoleh kesimpulan yaitu tingkat keberhasilan program transformasi Hough dalam mendeteksi dan merekonstruksi garis lurus pada suatu citra ada yang mencapai 100% dan ada pula yang hanya mencapai 20%. Hal ini dikarenakan garis lurus yang tidak terdeteksi memiliki besar  $R$  yang merupakan hasil pentransformasian ke dalam koordinat polar di bawah besar  $R$  maksimum yang telah dikalikan dengan nilai ambang, sehingga garis lurus tersebut tidak terdeteksi sebagai garis lurus. Hasil rata-rata persentase keberhasilan total adalah sebesar 90%.

**Kata-kunci:** aras keabuan, deteksi tepi, deteksi garis, rekonstruksi garis, transformasi Hough.

## I. PENDAHULUAN

### 1.1 Latar belakang

Aplikasi ilmu bidang teknik elektro dewasa ini semakin berkembang. Salah satunya dalam subbidang pengolahan citra digital. Sebagai contohnya adalah aplikasi deteksi garis lurus dalam suatu citra yang perlu dilakukan karena pada dasarnya bentuk suatu kurva atau bidang adalah suatu bentukan dari beberapa garis lurus.

Pada tugas akhir ini disusun sebuah program aplikasi transformasi Hough untuk deteksi garis lurus untuk menunjang penelitian mengenai pengenalan kurva atau bidang selanjutnya. Program komputer melakukan deteksi berdasarkan ciri khas dari citra yang diolah, yang kemudian digunakan sebagai patokan untuk mendeteksi citra yang dibaca.

### 1.2 Tujuan dan Manfaat

Tujuan dari tugas akhir ini adalah:

1. Membuat program yang dapat memunculkan dan mempertegas keberadaan garis lurus dalam sebuah citra berdasarkan ciri citra yang dimiliki.
2. Menganalisis hasil deteksi garis lurus yang diperoleh dari Transformasi Hough.

Manfaat dari tugas akhir ini adalah dapat dijadikan sebagai pijakan penelitian selanjutnya misalnya untuk deteksi kurva ataupun pengenalan objek.

### 1.3 Pembatasan Masalah

Hal-hal yang dibahas pada penulisan tugas akhir dibatasi pada:

1. Citra yang diolah adalah citra hasil proses pendigitalan, tanpa membahas proses pengambilan datanya.

2. Aplikasi transformasi Hough untuk deteksi garis lurus hanya sebatas mendeteksi adanya garis lurus dalam suatu citra.
3. Tugas akhir tidak membahas pengubahan citra berwarna menjadi citra aras keabuan secara rinci.
4. Tugas akhir tidak membahas tentang detektor Canny secara rinci.

## II. KAJIAN PUSTAKA

### 2.1 Pengertian Citra

Citra (*image*) secara harfiah diartikan sebagai gambar pada bidang dua dimensi (dwimatra). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi.

### 2.2 Derajat Keabuan

Intensitas cahaya pada titik  $(x,y)$  yang disimbolkan  $f(x,y)$ , disebut dengan derajat keabuan (*grey level*). Derajat keabuan berada dalam rentang antara hitam dan putih. Citra yang memiliki derajat keabuan dari hitam ke putih ini disebut citra hitam-putih (*greyscale image*) atau citra monokrom.

### 2.3 Digitalisasi Citra

Digitalisasi citra adalah sajian citra dari fungsi malar (kontinu/menerus) menjadi nilai-nilai diskret. Sedangkan citra digital adalah citra yang dibentuk oleh nilai-nilai diskret hasil digitalisasi. Citra digital juga lazim disajikan secara matriks berukuran  $N$  baris dan  $M$  kolom.

Masing-masing elemen dalam citra digital (dalam hal ini adalah elemen matriks) disebut *image element*, *picture element*, atau piksel atau

\* Mahasiswa Jurusan Teknik Elektro UNDIP

\*\* Dosen Jurusan Teknik elektro UNDIP

pel. Sehingga sebuah citra yang berukuran  $N \times M$  memiliki piksel sebanyak  $NM$  buah.

## 2.4 Deteksi Tepi

Deteksi tepi dilakukan untuk meningkatkan penampakan garis pada citra tepi. Terdapat beberapa macam metode untuk mendeteksi tepi. Pada penelitian ini digunakan metode Canny untuk mendeteksi tepi.

## 2.5 Transformasi Hough

Transformasi Hough menggunakan bentuk parametrik dan menggunakan pemungutan suara terbanyak (*voting*) untuk menentukan nilai parameter yang tepat. Apabila dalam citra terdapat beberapa garis yang saling berpotongan pada suatu titik, maka apabila kemudian titik tersebut ditransformasi ke dalam ruang parameter  $m - c$  akan didapati bahwa transformasi dalam ruang parameter  $m - c$  adalah sebuah garis lurus dengan persamaan garis dinyatakan sebagai,

$$y_i = mx_i + c \quad (2.1)$$

Sebaliknya jika dalam citra terdapat sebuah garis lurus, dimana garis tersebut mempunyai persamaan seperti persamaan (2.1), maka apabila ditransformasi ke dalam ruang parameter  $m - c$  akan diperoleh bahwa transformasi dalam ruang parameter  $m - c$  adalah beberapa garis yang saling berpotongan dalam suatu titik. Persamaan transformasi diperoleh dengan cara memanipulasi persamaan garisnya yaitu persamaan (2.1) menjadi bentuk parametrik menjadi persamaan.

$$c = y_i - mx_i \quad (2.2)$$

Perhitungan transformasi dilakukan dengan cara menghitung tiap titik dalam citra kedalam semua variasi nilai  $m - c$ .

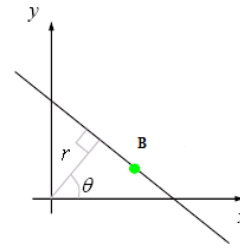
Dalam kenyataannya, apabila ditemui sebuah garis vertikal, maka akan terjadi masalah dalam penghitungannya dikarenakan garis vertikal mempunyai nilai gradien kemiringan  $m$  yang besarnya tak berhingga. Sebagai alternatifnya digunakan persamaan (2.3).

$$r = x \cos \theta + y \sin \theta \quad (2.3)$$

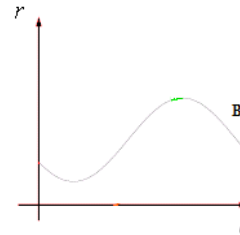
dimana gambar koordinat kartesiannya ditunjukkan pada Gambar 2.1.

Sebuah titik B dalam gambar tersebut apabila ditransformasi ke dalam ruang parameter  $r - \theta$  maka akan menjadi seperti Gambar 2.2.

Apabila di dalam citra terdapat suatu garis lurus, maka jika garis lurus citra ditransformasi kedalam ruang parameter  $r - \theta$  akan terjadi suatu titik penumpukan antar kurva sinusoida hasil pentransformasian masing-masing komponen piksel garis yang membentuk garis lurus tersebut.



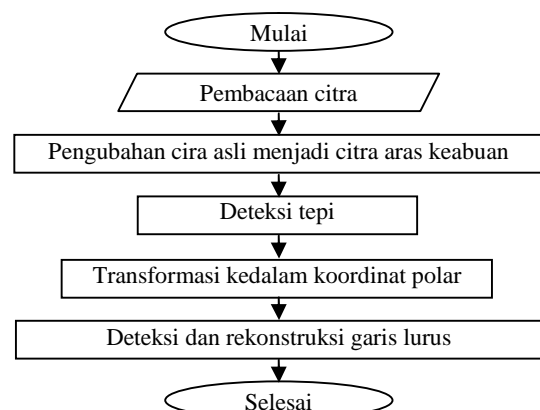
Gambar 2.1 Citra sebuah garis dalam koordinat Kartesian dengan satu titik uji.



Gambar 2.2 Hasil transformasi dari satu titik uji dalam ruang parameter  $r - \theta$ .

## III. PERANCANGAN PROGRAM

Pada bab ini dibahas perancangan perangkat lunak untuk aplikasi transformasi Hough untuk deteksi garis lurus. Pada perancangan ini digunakan Matlab 7.1 karena fungsi-fungsinya lengkap dan mudah dalam pembuatan tampilan antarmuka penggunaan grafis (GUI). Pada tugas akhir ini digunakan berkas citra dengan ekstensi **.bmp**, **.jpeg** atau **.jpg**. Secara garis besar perangkat lunak yang akan dirancang memiliki diagram alir seperti yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram alir program.

Urutan langkah-langkah proses jalannya program adalah sebagai berikut.

### 3.1 Pembacaan Berkas Citra

Pembacaan berkas citra dilakukan dengan perintah `imread`. Pembacaan berkas citra dilakukan dengan perintah Matlab sebagai berikut.

```
GbrAsli = imread(filename);
axes(handles.axGbrAsli);
```

```
imshow(GbrAsli);
```

### 3.2 Pengubahan Citra menjadi Citra Aras Keabuan

Setelah dilakukan pembacaan citra, dilanjutkan dengan pengubahan citra menjadi citra aras keabuan dengan menekan tombol **Aras Keabuan**. Pengubahan citra berwarna menjadi citra aras keabuan dalam program ini menggunakan perintah `rgb2gray`. Senarai programnya sebagai berikut.

```
gbrGR = rgb2gray(gbrA);
```

### 3.3 Deteksi Tepi

Setelah dilakukan pengubahan citra menjadi citra aras keabuan, lalu dilanjutkan dengan pendeteksian tepi dengan menekan tombol **Deteksi Tepi**. Deteksi tepi dilakukan menggunakan detektor Canny. Pendeteksian citra dengan detektor Canny dalam program ini menggunakan perintah `edge(gbrGR, 'canny')`. Senarai program sebagai berikut.

```
axes(handles.axGbTepi);
gbrDT = edge(gbrGR, 'canny');
imshow(gbrDT);
```

### 3.4 Transformasi Citra Tepi dari Koordinat Kartesian menjadi Koordinat Polar

Setelah dilakukan pendeteksian tepi, lalu dilanjutkan dengan transformasi dari koordinat Kartesian ke dalam koordinat polar dengan menekan tombol **Plot Hough**. Transformasi dari koordinat Kartesian ke dalam koordinat polar dilakukan dengan menggunakan fungsi **Radon** yang ada dalam Matlab 7.1. Senarai programnya sebagai berikut.

```
theta_step=1;
theta = 0:theta_step:360;
[R, xp] = radon(gbrDT, theta);

imagesc(theta, xp, R);
colormap(gray);
xlabel('\theta (degrees)');
ylabel('X\prime');
title('R_{\theta} (X\prime)');
colorbar
```

Fungsi Radon berfungsi untuk transformasi citra dari koordinat Kartesian ke dalam koordinat polar.

### 3.5 Deteksi dan Rekonstruksi Garis Lurus

Setelah dilakukan transformasi dari koordinat Kartesian ke dalam koordinat polar, lalu dilanjutkan dengan deteksi dan rekonstruksi garis lurus dengan menekan tombol **Garis**.

Tahap-tahap deteksi dan rekonstruksi garis lurus adalah sebagai berikut:

1. Penghitungan nilai maksimum dari R yaitu dengan perintah sebagai berikut.

```
max_R=max(max(R));
```

2. Penentuan nilai ambang sebesar 0,75 yaitu dengan perintah sebagai berikut.

```
threshold=.75;
```

3. Pencarian nilai R yang memenuhi ketentuan syarat nilai ambang, lalu membuat indeksnya dengan perintah sebagai berikut.

```
[II,JJ]=find( R >= (max_R
*threshold));
```

4. Rekonstruksi garis yang memenuhi ketentuan syarat nilai ambang dengan perintah sebagai berikut.

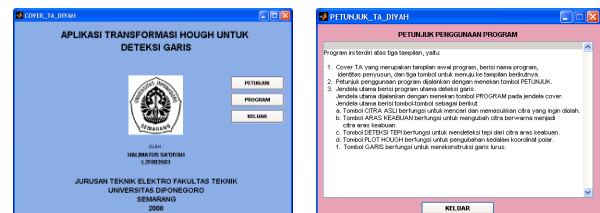
```
[line_n, d]=size(II);
imshow(gbrA)
for k=1:line_n
    j=JJ(k);
    i=II(k);
    R_i=(j-1)*theta_step;
    xp_i=xp(i);
    [n,m]=size(gbrDT);
    x_origin=m/2
        +(xp_i)*cos(R_i*pi/180);
    y_origin=n/2
        -(xp_i)*sin(R_i*pi/180);

    x1=1;
    xe=m;
    y1=(y_origin-(x1-
        x_origin)*tan((R_i)-
        90)*pi/180));
    ye=(y_origin-(xe-
        x_origin)*tan((R_i)-
        90)*pi/180));
    xv=[x1 xe];
    yv=[y1 ye];
    hold on
    line(xv,yv)
    plot(x_origin,y_origin,'g')
end
```

## IV. HASIL PENELITIAN DAN PEMBAHASAN

### 4.1 Tampilan Program Aplikasi Transformasi Hough Untuk Deteksi Garis

Sebelum dibahas mengenai Hasil pengujian, diperlihatkan terlebih dahulu tampilan dari program aplikasi transformasi Hough untuk deteksi garis sebagai berikut.



Gambar 4.1 Tampilan program.

(a) Tampilan awal program.

(b) Tampilan petunjuk penggunaan program.

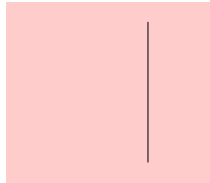


Gambar 4.2 Contoh tampilan program utama.

## 4.2 Contoh Pengujian Program Aplikasi Transformasi Hough Untuk Deteksi Garis Lurus

### 4.2.1 Pembacaan Citra

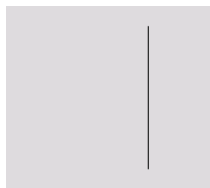
Pembacaan citra asli dan penampilan citra dengan cara menekan tombol **CITRA ASLI** pada program utama yang diperlihatkan pada Gambar 4.2, lalu dipilih citra yang diinginkan. Pada contoh ini dipilih citra **vertikal.bmp**. Hasil proses ini diperlihatkan pada Gambar 4.3.



Gambar 4.3 Citra uji **vertikal.bmp** asli.

### 4.2.2 Pengubahan Citra menjadi Citra Aras Keabuan

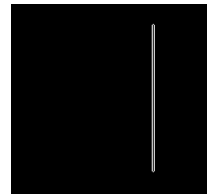
Pengubahan citra asli yaitu citra berwarna menjadi citra aras keabuan dilakukan dengan cara menekan tombol **ARAS KEABUAN** pada program utama yang diperlihatkan pada Gambar 4.2. Hasil proses ini diperlihatkan pada Gambar 4.4.



Gambar 4.4 Citra uji **vertikal.bmp** aras keabuan.

### 4.2.3 Deteksi Tepi

Deteksi tepi citra aras keabuan dengan cara menekan tombol **DETEKSI TEPI** pada program utama yang diperlihatkan pada Gambar 4.2. Hasil proses ini diperlihatkan pada Gambar 4.5. Proses deteksi tepi bertujuan untuk meningkatkan penampakan tepi pada citra. Suatu objek dapat dengan mudah dideteksi pada suatu citra jika objek cukup kontras dari latar belakangnya. Pada proses deteksi tepi ini hasilnya adalah berupa citra biner yang hanya memiliki dua nilai aras keabuan saja yaitu hitam dan putih, dimana aras keabuan putih menunjukkan batas tepi.



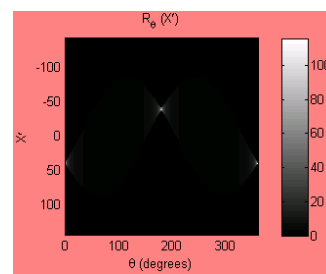
Gambar 4.5 Citra uji **vertikal.bmp** terdeteksi tepi.

### 4.2.4 Transformasi Citra Tepi dari Koordinat Kartesian ke dalam Koordinat Polar

Pentransformasian citra tepi dari koordinat Kartesian ke dalam koordinat polar dilakukan dengan cara menekan tombol **PLOT HOUGH** pada program utama yang diperlihatkan pada Gambar 4.2. Proses transformasi dari koordinat Kartesian ke dalam koordinat polar dilakukan pada tiap piksel citra batas tepi yang memiliki aras keabuan putih. Transformasi ini dilakukan dengan menggunakan persamaan.

$$r = x \cos \theta + y \sin \theta \quad (4.1)$$

Pada program ini, transformasi dari koordinat Kartesian ke dalam koordinat polar menggunakan fungsi Radon yang sudah dimiliki oleh Matlab 7.1. Hasil proses ini diperlihatkan pada Gambar 4.6.



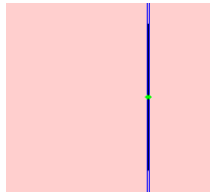
Gambar 4.6 Hasil pengubahan dari koordinat Kartesian menjadi koordinat polar citra uji **vertikal.bmp** terdeteksi tepi.

Pada proses ini diperlihatkan bahwa antar kurva sinusoida hasil transformasi dari koordinat Kartesian ke dalam koordinat polar yang dilakukan pada tiap piksel citra batas tepi terjadi beberapa titik penumpukan. Titik penumpukan disebabkan oleh beberapa kurva sinusoida hasil transformasi masing-masing komponen piksel citra batas tepi melewati titik penumpukan tersebut. Titik penumpukan ini menunjukkan adanya indikasi bahwa antar piksel citra batas tepi yang hasil transformasinya melewati titik penumpukan tersebut dapat ditarik suatu garis lurus. Semakin banyak kurva sinusoida hasil transformasi masing-masing komponen piksel citra batas tepi melewati titik penumpukan tersebut, maka semakin besar pula nilai penumpukan pada suatu titik. Semakin besar nilai penumpukan pada suatu titik, maka semakin besar pula kemungkinan bahwa antar piksel citra batas tepi tersebut dapat ditarik suatu garis lurus. Besarnya  $R$  menunjukkan besarnya nilai

penumpukan antar kurva sinusoida hasil transformasi masing-masing komponen piksel citra batas tepi.

#### 4.2.5 Deteksi dan Rekonstruksi Garis Lurus

Proses selanjutnya adalah deteksi dan rekonstruksi garis lurus. Deteksi garis lurus dilakukan dengan perbandingan antara besar tiap-tiap  $R$  yang ada terhadap besar  $R$  maksimum yang telah dikalikan dengan nilai ambang yang telah ditentukan yaitu sebesar 0,75. Besarnya  $R$  maksimum yang telah dikalikan dengan nilai ambang dijadikan sebagai batas minimal besar  $R$  yang harus dipenuhi sebagai syarat dideteksi dan direkonstruksi sebagai sebuah garis lurus. Deteksi dan rekonstruksi garis lurus dijalankan dengan cara menekan tombol **GARIS** pada program utama yang diperlihatkan pada Gambar 4.2. Hasil proses ini diperlihatkan pada Gambar 4.7.



Gambar 4.7 Citra uji **vertikal.bmp** terekonstruksi garis lurus.

Perekonstruksian garis lurus pada program ini dilakukan pada tiap-tiap titik penumpukan yang telah dideteksi sebagai garis lurus. Perekonstruksian garis lurus pada program ini dimulai dengan penghitungan dan penentuan letak titik asal terhadap titik tengah citra hasil pendeteksian tepi yang dijadikan sebagai titik referensi. Jarak antara letak titik asal terhadap letak titik tengah adalah merupakan jarak terdekat antara garis lurus dengan titik tengah yang dalam persamaan (4.1) jarak tersebut dinyatakan dengan symbol  $r$ . Titik tengah citra adalah suatu titik yang berada persis di tengah matriks suatu citra. Penentuan letak titik asal terhadap titik tengah citra yang dijadikan sebagai titik referensi dijadikan sebagai alat bantu untuk mempermudah proses perekonstruksian selanjutnya. Setelah itu ditentukan batas titik  $x$  sesuai batas kolom matriks pada citra asli. Selanjutnya dihitung dan ditentukan batas titik  $y$  dengan mengacu pada letak titik asal dan batas titik  $x$ . Setelah itu direkonstruksi garis lurus dan titik asalnya sesuai dengan batas titik  $x$  dan batas titik  $y$  yang bersesuaian yang telah dihitung sebelumnya. Titik asal direkonstruksi dengan warna hijau, sedangkan garis direkonstruksi dengan warna biru. Warna biru dan corak garis tersebut otomatis ditentukan oleh Matlab 7.1.

#### 4.3 Hasil Pengujian Program Aplikasi Transformasi Hough Untuk Deteksi Garis Lurus

Berikut ini hasil pengujian masing-masing citra uji yang lainnya.

TABEL 4.1 HASIL PENGUJIAN CITRA UJI

Nama citra uji	citra asli	Rekonstruksi garis
horizontal.bmp		
vertikal.bmp		
miring.bmp		
miring2.bmp		
kotak.bmp		
segi5.bmp		
segi6.bmp		
segi_tiga.jpeg		

Berdasarkan pengujian terhadap citra uji yang sudah dilakukan, dapat dilakukan penilaian tingkat keberhasilan program yang disajikan dalam Tabel 4.2. Penilaian ini berdasarkan hasil visualisasi yang tampak dari citra asli dan rekonstruksi garisnya. Beberapa rekonstruksi garis yang saling berimpitan dianggap sebagai sebuah garis.



TABEL 4.2 PENILAIAN TINGKAT KEBERHASILAN PROGRAM  
MENGUJI CITRA UJI.

Nama citra	Jumlah deteksi	Jumlah garis yang ada	% keberhasilan
horisontal.bmp	1 buah garis	1 buah garis	100%
vertikal.bmp	1 buah garis	1 buah garis	100%
miring.bmp	1 buah garis	1 buah garis	100%
miring2.bmp	1 buah garis	1 buah garis	100%
kotak.bmp	4 buah garis	4 buah garis	100%
segi5.bmp	5 buah garis	5 buah garis	100%
segi6.bmp	6 buah garis	6 buah garis	100%
segi_tiga.jpeg	1 buah garis	5 buah garis	20%

Berdasarkan Tabel 4.2 dapat dilihat bahwa tingkat keberhasilan program transformasi Hough yang dibuat ada yang mencapai 100% dan ada pula yang hanya mencapai 20%. Hal ini dikarenakan garis lurus yang tidak terdeteksi memiliki besar R yang merupakan hasil transformasi ke dalam koordinat polar di bawah besar R maksimum yang telah dikalikan dengan nilai ambang, sehingga garis lurus tersebut tidak terdeteksi sebagai garis lurus. Hasil rata-rata persentase keberhasilan berdasarkan Tabel 4.8 adalah sebesar 90%.

## V. PENUTUP

### 5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian yang telah dilakukan adalah sebagai berikut.

1. Program yang dibuat dengan Matlab7.1 digunakan untuk mendeteksi garis lurus menggunakan transformasi Hough.
2. Tingkat keberhasilan program transformasi Hough dalam mendeteksi garis lurus dalam citra uji yang diberikan ada yang mencapai 100% dan ada pula yang hanya mencapai 20%.
3. Tingkat keberhasilan program transformasi Hough dalam mendeteksi ada yang hanya mencapai 20% dikarenakan oleh garis lurus yang tidak terdeteksi tersebut memiliki besar R yang merupakan hasil transformasi ke dalam koordinat polar berada di bawah besar R maksimum yang telah dikalikan dengan nilai ambang, sehingga garis lurus tersebut tidak terdeteksi sebagai garis lurus.
4. Rata-rata persentase keberhasilan dari citra uji keseluruhan mencapai 90%.

### 5.2 Saran

Penelitian lebih lanjut diharapkan dapat memperbaiki kekurangan yang ada dan diharapkan dapat mengembangkan apa yang telah dilakukan pada penelitian ini. Untuk itu disarankan hal-hal berikut.

1. Perlu penelitian untuk pengembangan perangkat lunak transformasi Hough menggunakan bahasa-bahasa pemrograman selain Matlab, misalnya C++ dan Visual Basic. Selanjutnya, dibandingkan kinerja masing-masing program dilihat dari sisi waktu pengolahan maupun penggunaan media penyimpan (memori). Dengan demikian nantinya dapat diketahui bahasa pemrograman yang paling sesuai.
2. Perlu dilakukan penelitian lanjutan tentang transformasi Hough untuk tujuan mendeteksi bentuk seperti lingkaran, elips, persegi panjang, dan lain sebagainya.

### DAFTAR PUSTAKA

- [1] Deipara, G. D., *Identifikasi Dini Kangker Rahim Menggunakan Metode Segmentasi Berdasar Deteksi Tepi*, Skripsi S-1, Universitas Diponegoro, Semarang, 2004.
- [2] Edyanto, J., *Matlab Bahasa Komputasi Teknis*, Andi Offset, Yogyakarta, 2000.
- [3] Hamarneh, G., K. Althaff, R. Abu-Gharbieh, *Automatic Line Detection*, [http://www.cs.sfu.ca/~hamarneh/ecopy/compvis1999\\_hough.pdf](http://www.cs.sfu.ca/~hamarneh/ecopy/compvis1999_hough.pdf), Maret 2008.
- [4] Jain, A.K., *Fundamental of Digital Image Processing*, Prentice Hall, New Jersey, 1989.
- [5] Munir, R., *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Informatika, Bandung, 2004.
- [6] Paulus, E., Y. Nataliani, *GUI Matlab*, Andi Offset, Yogyakarta, 2007.
- [7] Putriningsih, D., *Identifikasi Kelebihan Kolesterol Berdasarkan Pengamatan Citra Iris Mata*, Skripsi S-1, Universitas Diponegoro, Semarang, 2007.
- [8] Ramza, H., Y. Dewanto, *Teknik Pemrograman Menggunakan Matlab*, Grasindo, Jakarta, 2007.
- [9] Sugiharto, A., *Pemrograman GUI dengan Matlab*, Andi Offset, Yogyakarta, 2006.
- [10] Wahyuni, T. B., *Deteksi Osteoporosis dan Tumor Tulang Menggunakan Segmentasi Berdasar Intensitas Citra*, Skripsi S-1, Universitas Diponegoro, Semarang, 2005.
- [11] Yudisthira, F., *Pengukuran Kerja Detektor-Detektor Tepi Citra dengan Acuan Detektor Canny*, Skripsi S-1, Universitas Diponegoro, Semarang, 2004.



Halimatus Sa'diyah  
(L2F003503) lahir di Solo.  
Menjalani pendidikan di SDN  
Semanggi Lor 127 Surakarta,  
SMP Al-Islam 1 Surakarta,  
SMA Al-Islam 1 Surakarta.  
Saat ini sedang menyelesaikan  
pendidikan S1 di Universitas  
Diponegoro Jurusan Teknik

Menyetujui dan Mengesahkan

Pembimbing I

R. Rizal Isnanto, S.T., M.M., M.T.  
NIP. 132 288 515  
Tanggal.....

Pembimbing II

Achmad Hidayatno, S.T., M.T.  
NIP. 132 137 933  
Tanggal.....